Least-Squares Fitting

James R. Graham
2014/9/21

A straight line fit

Suppose that we have a set of *N* observations (x_i, y_i) where we believe that the measured value, y, depends linearly on x, i.e.,

$$y = mx + c.$$

For example, suppose a body is moving with constant velocity what is the speed (*m*) and initial (*c*) position of the object?

Given our data, what is the best estimate of *m* and *c*? Assume that the independent variable, x_i, is known exactly, and the dependent variable, y_i, is drawn from a Gaussian probability distribution function with constant standard deviation, $\sigma_i = const$. Under these circumstances the most likely values of *m* and *c* are those corresponding to the straight line with the total minimum square deviation, i.e., the quantity

$$\chi^2 = \sum_i \left[y_i - (mx_i + c) \right]^2$$

is minimized when *m* and *c* have their most likely values. Figure 1 shows a typical deviation.

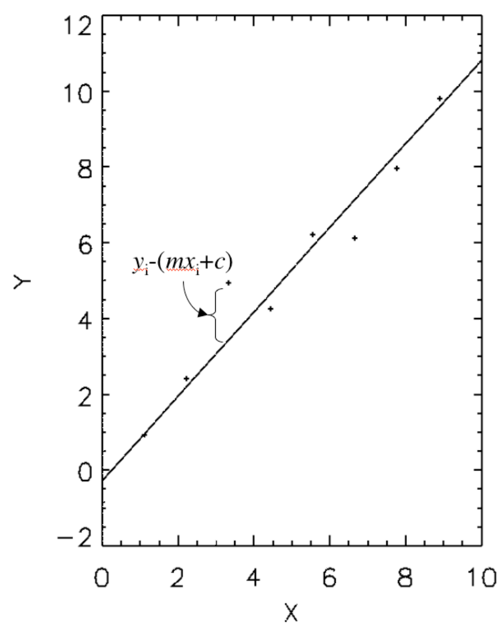

Figure 1: Some data with a least squares fit to a straight line. A typical deviation is illustrated.

The best values of m and c are found by solving the simultaneous equations,

$$\frac{\partial}{\partial m}\chi^2 = 0, \quad \frac{\partial}{\partial c}\chi^2 = 0.$$

Evaluating the derivatives yields

$$\frac{\partial}{\partial m}\chi^2 = \frac{\partial}{\partial m}\sum_i \left[y_i - (mx_i + c)\right]^2 = 2m\sum_i x_i^2 + 2c\sum_i x_i - 2\sum_i x_i y_i = 0$$
$$\frac{\partial}{\partial c}\chi^2 = \frac{\partial}{\partial c}\sum_i \left[y_i - (mx_i + c)\right]^2 = 2m\sum_i x_i + 2cN - 2\sum_i y_i = 0.$$

Which can conveniently be expressed in matrix form,

$$\begin{pmatrix} \sum x_i^2 & \sum x_i \\ \sum x_i & N \end{pmatrix}\begin{pmatrix} m \\ c \end{pmatrix} = \begin{pmatrix} \sum x_i y_i \\ \sum y_i \end{pmatrix}$$

and solved by multiplying both sides by the inverse,

$$\begin{pmatrix} m \\ c \end{pmatrix} = \begin{pmatrix} \sum x_i^2 & \sum x_i \\ \sum x_i & N \end{pmatrix}^{-1}\begin{pmatrix} \sum x_i y_i \\ \sum y_i \end{pmatrix}.$$

The inverse can be computed analytically, or in Python it is trivial to compute the inverse numerically, as follows.

Example Python

```
# Test least squares fitting with simulated data.
import numpy as np
import matplotlib.pyplot as plt

nx = 20             # Number of data points
m = 1.0             # Gradient
c = 0.0             # Intercept

x = np.arange(nx,dtype=float)       # Independent variable
y = m * x + c                       # dependent variable

# Generate Gaussian errors
sigma = 1.0                         # Measurement error

np.random.seed(1)                   # init random no. generator
errors = sigma*np.random.randn(nx)  # Gaussian distributed errors
ye = y + errors                     # Add the noise

plt.plot(x,ye,'o',label='data')
plt.xlabel('x')
plt.ylabel('y')

# Construct the matrices
ma = np.array([ [np.sum(x**2), np.sum(x)],[np.sum(x), nx ] ]  )
mc = np.array([ [np.sum(x*ye)],[np.sum(ye)]])
```

```
# Compute the gradient and intercept
mai = np.linalg.inv(ma)
print 'Test matrix inversion gives identity',np.dot(mai,ma)
md = np.dot(mai,mc)     # matrix multiply is dot

# Overplot the best fit
mfit = md[0,0]
cfit = md[1,0]
plt.plot(x, mfit*x + cfit)
plt.axis('scaled')
plt.text(5,15,'m = {:.3f}\nc = {:.3f}'.format(mfit,cfit))
plt.savefig('lsq1.png')
```

See Figure 2 for the output of this program.

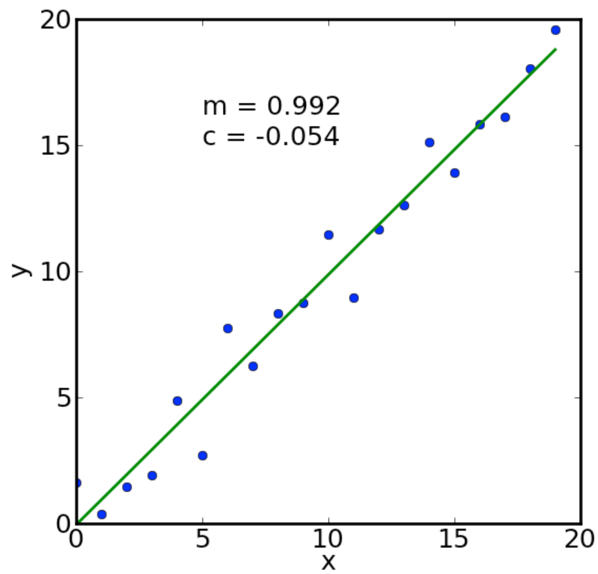

Figure 2—Least squares straight line fit. The true values are *m* = 1 and *c* = 0.

Error propagation

What are the uncertainties in the slope and the intercept? To begin the process of error propagation we need the inverse

$$\begin{pmatrix} \sum x_i^2 & \sum x_i \\ \sum x_i & N \end{pmatrix}^{-1} = \begin{pmatrix} N \Big/ \left[N\sum x_i^2 - \left(\sum x_i\right)^2 \right] & \sum x_i \Big/ \left[N\sum x_i^2 - \left(\sum x_i\right)^2 \right] \\ \sum x_i \Big/ \left[\left(\sum x_i\right)^2 - N\sum x_i^2 \right] & \sum x_i \Big/ \left[N\sum x_i^2 - \left(\sum x_i\right)^2 \right] \end{pmatrix},$$

so that we can compute analytic expressions for m and c,

$$\begin{pmatrix} m \\ c \end{pmatrix} = \begin{pmatrix} \sum x_i^2 & \sum x_i \\ \sum x_i & N \end{pmatrix}^{-1} \begin{pmatrix} \sum x_i y_i \\ \sum y_i \end{pmatrix} = \begin{pmatrix} \dfrac{\sum x_i \sum y_i - N \sum x_i y_i}{\left(\sum x_i\right)^2 - N \sum x_i^2} \\ \dfrac{\sum x_i \sum x_i y_i - \sum x_i^2 \sum y_i}{\left(\sum x_i\right)^2 - N \sum x_i^2} \end{pmatrix}.$$

The results of error propagation show that if $z = z(x_1, x_2, .. x_N)$

$$\sigma_z^2 = \sum_i \left(\partial z / \partial x_i\right)^2 \sigma_i^2 ,$$

assuming uncorrelated data (i.e., zero covariance). Thus,

$$\sigma_m^2 = \sum_j \left(\partial m / \partial y_j\right)^2 \sigma_j^2 \quad \text{and} \quad \sigma_c^2 = \sum_j \left(\partial c / \partial y_j\right)^2 \sigma_j^2 .$$

The expression for the derivative of the gradient, m, is

$$\frac{\partial m}{\partial y_j} = \frac{\partial}{\partial y_j} \left(\frac{\sum x_i \sum y_i - N \sum x_i y_i}{\left(\sum x_i\right)^2 - N \sum x_i^2} \right) = \frac{\sum x_i - N x_j}{\left(\sum x_i\right)^2 - N \sum x_i^2}$$

because $\left(\partial y_i / \partial y_j\right) = \delta_{ij}$. If we assume that the measurement error is the same for each point then

$$\begin{aligned} \sigma_m^2 &= \sigma^2 \sum_j \left(\frac{\sum x_i - N x_j}{\left(\sum x_i\right)^2 - N \sum x_i^2} \right)^2 \\ &= \frac{\sigma^2}{\left[\left(\sum x_i\right)^2 - N \sum x_i^2\right]^2} \sum_j \left[\left(\sum x_i\right)^2 - 2 N x_j \sum x_i + N^2 x_j^2 \right] \\ &= \frac{\sigma^2}{\left[\left(\sum x_i\right)^2 - N \sum x_i^2\right]^2} \left[N \left(\sum x_i\right)^2 - 2N \left(\sum x_i\right)^2 + N^2 \sum x_i^2 \right] \\ &= \frac{N \sigma^2}{N \sum x_i^2 - \left(\sum x_i\right)^2} \end{aligned}$$

Similarly,

$$\frac{\partial c}{\partial y_j} = \frac{\partial}{\partial y_j} \left(\frac{\sum x_i \sum x_i y_i - \sum x_i^2 \sum y_i}{\left(\sum x_i\right)^2 - N \sum x_i^2} \right) = \frac{x_j \sum x_i - \sum x_i^2}{\left(\sum x_i\right)^2 - N \sum x_i^2}$$

and

$$\begin{aligned}
\sigma_c^2 &= \sigma^2 \sum_j \left(\frac{x_j \sum x_i - \sum x_i^2}{\left(\sum x_i\right)^2 - N\sum x_i^2} \right)^2 \\
&= \frac{\sigma^2}{\left[\left(\sum x_i\right)^2 - N\sum x_i^2\right]^2} \sum_j \left[x_j^2 \left(\sum x_i\right)^2 - 2x_j \sum x_i \sum x_i^2 + \left(\sum x_i^2\right)^2 \right] \\
&= \frac{\sigma^2 \sum x_i^2}{\left[\left(\sum x_i\right)^2 - N\sum x_i^2\right]^2} \left[N\sum x_i^2 - \left(\sum x_i\right)^2 \right] \\
&= \frac{\sigma^2}{\left[\left(\sum x_i\right)^2 - N\sum x_i^2\right]^2} \left[\sum x_i^2 \sum x_i - 2\sum x_i \sum x_i^2 + N\sum x_i^2 \right] \\
&= \frac{\sigma^2 \sum x_i^2}{N\sum x_i^2 - \left(\sum x_i\right)^2}.
\end{aligned}$$

If we do not know, *a priori*, the standard deviation of the measurements, σ, the best estimate is derived from the deviations from comparing the data to the fit, i.e.,

$$\sigma^2 = \frac{1}{N-2} \sum_i \left[y_i - (mx_i + c) \right]^2.$$